A JPL Information Architecture Technical Concept White Paper

Architecture and System Engineering Office
Institutional Computing and Information Services

J. Steven Hughes
Daniel Crichton

Jet Propulsion Laboratory

D-23710

January 2002

Table of Contents

1. Introduction 3
2. The JPL Data Space 4
   2.1 Data Models 4
   2.2 Data Formats 8
   2.3 Complexity of Data Models 8
3. The JPL Name Space 11
4. The JPL Resource Space 12
5. System Modeling for Data Architectures and Engineering 14
6. Data Policies and Standards 16
7. Data Processes and Functions 18
   7.1 Capture 18
   7.2 Locate 18
   7.3 Access 18
   7.4 Exchange 18
   7.5 Interpret 19
   7.6 Distribute 19
8. Data Management Services 19
   8.1 Database Hosting Service 20
   8.2 Metadata Service 20
   8.3 Catalog and Archive Service 20
   8.4 Product Exchange Service 21
   8.5 Data Distribution Service 21
9. Benefits to JPL 22
10. Plans and Recommendations 27
11. References 29

Copyright Jet Propulsion Laboratory, California Institute of Technology

1. Introduction

Information is critical to enabling JPL to meet its strategic mission. At JPL, data is distributed across the enterprise and is often captured and managed independently without reference to any standard information architectural principles. Interoperability and efficient search and retrieval of data products across these systems is difficult because users are required to connect to each individual data system and deal with dissimilar and often unfamiliar organizing principles. It also makes the development of software systems that work across organization and project boundaries challenging if the organizing principles are not explicitly defined.

Effective search, retrieval and exchange of data products within and across these data systems depend crucially on information architectures that are organized on principles known to the prospective users and designed to reflect relevant relationships. Well-designed information architectures enable data browsing, data mining, and correlative studies across data sets.

IEEE's recommendation for architectural description [1] states, "the application of architectural concepts is increasingly being recognized as a critical element in the successful development and evolution of software systems. Experience increasingly shows that software systems developed using architectural concepts are more often successful and turn out 'faster, better, and cheaper.' As a result there now exists a growing body of knowledge in the application of architectural concepts to software systems to attain the benefits of reduced cost and increased quality, such as usability, flexibility, reliability, and interoperability."

Architectural concepts for software systems are clearly applicable to information systems. With the advent of the Web and the move toward greater information system interoperability, the requirement to share and correlate information has only increase the need for well designed information systems. For example, institutional decision support and enterprise knowledge management systems increasingly require access to institutional databases across many disparate domains. The development of architectural concepts and the incorporation of architectural thinking into information systems development will be critical to the success of these efforts.

At JPL, we envision an information architecture that identifies, documents, and promotes standard architectural concepts for building information systems across JPL institutional boundaries. It defines standard processes for capturing, locating, accessing, interpreting, exchanging, and distributing information across JPL's diverse system infrastructure. It supports existing, heterogeneous information repositories and promotes faster, better,

Copyright Jet Propulsion Laboratory, California Institute of Technology

cheaper, and more homogeneous future information system development. It addresses cost, risk, schedule, and security supporting JPL strategic goals.

This white paper outlines the JPL information architecture concept. It discusses JPL's information space as a model for describing and structuring the JPL information architecture. It breaks the information architecture into logical structures encompassed by the data, name and resource spaces, and defines key processes that enable the management of data in each space. It also discusses the benefits to JPL. Finally, it recommends data management services that implement the information architecture.

2. The JPL Data Space

The key to enabling interoperability across distributed information systems is the ability to understand the organizing principles of each of the component systems. We will define data space to be the scope over which a single set of organizing principles applies. In general, a data space will uniquely identify a set of data resources so that there is no ambiguity when resources are combined to achieve interoperability.

A data space can be loosely described as a domain specific collection of data and its meaning, relationships, and behavior. A domain, such as planetary science, sets the boundaries of the data to be considered and typically provides the means to collect, describe, organize, store, and interpret the domain data. Other examples of data spaces include the earth science, JPL administrative, and vendor manufacturing data collections.

2.1 Data Models

The key organizing principle within a data space is the data model. A data model is used to formally describe the data in terms of its structure and relationships. Examples of data models include the Planetary Science data model [2] for the Planetary Data System (PDS) data archive, the STEP model [3] for manufacturing, and the Dublin Core [4] recommended data elements for describing electronic resources on the Web.

2.1.1 Metamodels

A metamodel is used to describe a data model. Often ambiguously also called "model", a metamodel prescribes or provides the syntax and semantics for creating and documenting data models as illustrated in Figure 1. An example of a metamodel is the Entity-Relationship (E-R) model. Metamodels provide a set of normative or prescribing standards for capturing the meaning, relationships and behavior of data within the data space. They often use graphical notation to define the objects and relationships that exist within the data space.

Copyright Jet Propulsion Laboratory, California Institute of Technology

Meta Model, Data Models, and Domains

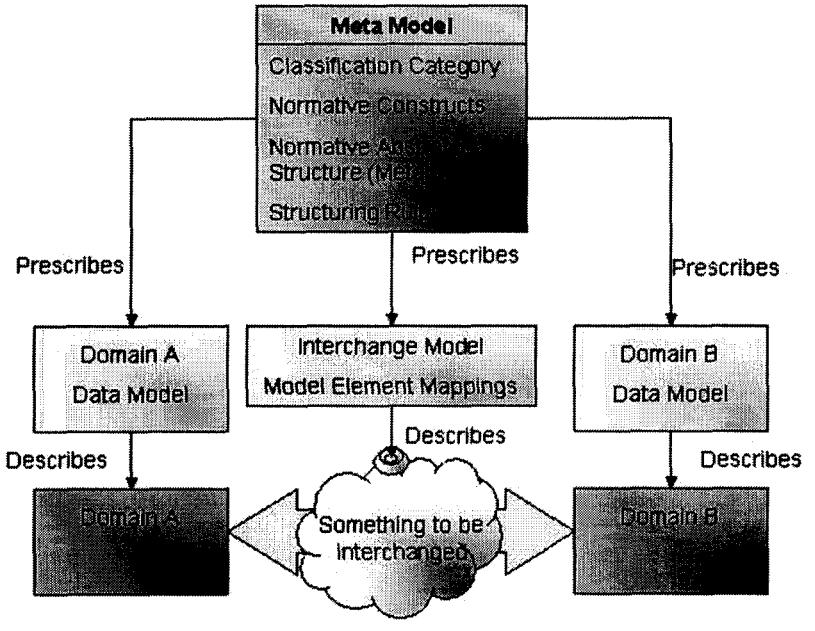

Figure 1 – Metamodel, Data Models, and Domain Relationships.

In the example illustrated in Figure 2, the PDS used the E-R metamodel to define how images of a planet are related to the spacecraft instrument that captured them. In the PDS data model, an entity called "data set" was first defined to describe a collection of images and then an "instrument" entity was defined to provide both summary and detailed descriptions about an instrument's functionality. A one-to-many relationship, "produces," was then defined between the instrument and data set entities to allow one instrument to be related to the many data sets that it could produce.

Copyright Jet Propulsion Laboratory, California Institute of Technology

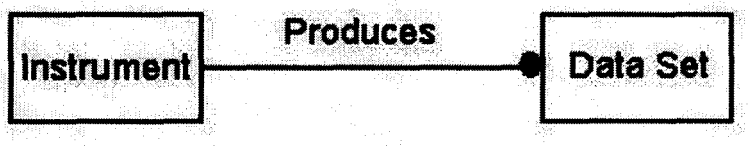

Figure 2 – PDS E-R Data Model for Data Set and Instrument

The Entity-Relationship (E-R) model is only one of many metamodels. Metamodels can be classified as either object-based or record-based. The object-based metamodels include the E-R and object-oriented metamodels where the record-based include the relational, networked, and hierarchical data models. Each of these metamodels has its own syntax, semantic constructs, and best practices for their application within information systems. Within JPL, each of these has probably been used at least once and has probably been applied with varying degrees of compliance to the specifications. In fact, only the E-R and relational metamodels come close to having what can be considered well-defined and widely accepted specifications.

2.1.2 Search and Retrieval

To allow efficient search and retrieval of data products within an information system, the user is required to understand the underlying data model. To achieve this, information system developers typically rely on a user interface to conceptually represent the data model in a user friendly and intuitive manner. For example, because planetary science data sets are related to target bodies, an intuitive user interface might start with a request for a target body, possibly represented by an icon depicting a planet or satellite. The user would then be guided to other relevant entities of the model using forms interfaces or image maps. In other words, the user interface simply represents to the user the underlying organizing principles as described in the information system data model. For an information system with a well-defined data model, the development and use of the user interface is relatively simple.

Search and retrieval across more than one information system is not much more difficult if the information systems belong to the same domain and use the same data model. However, this is seldom the case. Often the need arises to search across multiple domains that have little in common except that the user believes that a relationship exists. For example, even though the Hubble space telescope is considered part of the astrophysics domain—focusing on objects outside our solar system—the telescope is often used to

Copyright Jet Propulsion Laboratory, California Institute of Technology

capture images of planets and their satellites. So it is conceivable for a planetary scientist to request a search across both the astrophysics and planetary domains for images of Jupiter. However, the data models of the two domains are entirely different. The astrophysics domain works primarily with extra-solar objects and assumes stationary target bodies that have the celestial sphere for a frame of reference. It also allows target bodies to have many names. The planetary domain, however, deals with orbiting target bodies, fly-by spacecraft, and instruments on moveable scan arms, and so must deal with many related frames of reference. In addition, each target body is allowed only one standard name. So, a single query to search for the images of Jupiter across the information systems in both domains requires collecting all the identifiers used for the planet Jupiter. Finding an object by location in both domains is currently impossible using a single query.

2.1.3 Metadata Registries

Any solution to this interoperability problem involves identifying and describing commonalities across domain data models. This typically requires that the individual data models be normalized at the metamodel level, for example expressing the data models using the relational metamodel. This can and has been done manually with difficulty. However, we propose to design a metadata registry using a standard and comprehensive metamodel into which domain data models can be imported. Once one or more data models have been imported, domain experts will be able to analyze them within a common syntactic and semantic structure. The following scenario illustrates the approach. After importing the astrophysics and planetary data models into the metadata registry, the astrophysics and planetary terms for target bodies would be identified as two terms for the same concept. They could then be described as synonyms since for example there is only a simple syntactical difference between TARGET_NAME= JUPITER and OBJECT_ID=JUP.

More significant semantic differences would be handled using ontological concepts. For example both the planet Jupiter and the star Vega are target bodies but are targeted for study in different frames of reference. To capture the similarity that both can be target bodies of interest, a common ontological concept is first defined and the individual domain entities are then related to it. The domain specific frame of reference information remains within the domain data models. The resulting relationships span two domains and enables interoperability and the exchange of information.

The immediate question, however, is what metamodel should be used for the metadata registry. For example, the planetary domain data model was developed using the E-R model and implemented using the Object Description Language (ODL), a language developed and used primarily within the planetary science community. The astrophysics domain has an implicit data model based on a common standard data format. Neither community would readily adopt the other's data modeling methodology. This situation calls for the adoption or development of a reference data modeling methodology. The two communities could continue to use their own metamodels internally but would agree to

Copyright Jet Propulsion Laboratory, California Institute of Technology

import their data models into the metadata registry after expressing their data models using the reference metamodel.

It is proposed that a reference metamodel be adopted or developed based upon a review of the standard object-based and record-based metamodels and the Universal Modeling Language (UML) [5]. It is also proposed that the eXtensible Markup Language (XML) [6] [7] be used as the common syntax due to its near universal acceptance and flexibility of application. The reference metamodel will provide the syntax and semantics for creating and documenting the metadata registry data model. A key requirement for this reference metamodel is that it be comprehensive enough to subsume all metamodels that were used to define the data models to be imported into the registry. It should also specify the processes for applying the metamodel and subsequent validation and maintenance.

2.2 Data Formats

Data formats are used to describe the structure of a data granule, a unit of data that is independently managed. For example within the space sciences a camera image is a data granule and the Flexible Image Transport System (FITS), Hierarchical Data Format (HDF), and Common Data Format (CDF) are data formats.

In a data model, data granules are typically described conceptually and related to other objects in the domain. For example, an image might have attributes such as identifier, exposure duration, filter number, target body, and data format. However, the data format attribute simply identifies the data format used. The data format is itself a data model at the data representation level. If for example, FITS were specified as the data format for an image, a user would reference the FITS specification for a description of the data structure used to store the image.

For the information architecture it is proposed to review data formats available for space sciences and make recommendations for their use. For example, the eXtensible Data Format (XDF) [8], a new, general purpose, science data format based on XML, will be considered.

2.3 Data Model Scope

Enterprise data models are developed for varying purposes and with varying degrees of formalization, and scope. They range from simply classifying entities within a domain to

Copyright Jet Propulsion Laboratory, California Institute of Technology

fully describing the entities and their behavior. The following list of commonly used data models illustrates this range.

A **taxonomy** is a classification scheme. Within a domain, a taxonomy classifies domain entities according to a set of distinguishing characteristics and is typically one of the first organizational principles applied while data modeling. Sometimes it is the only. A classic taxonomy is the kingdom, phylum hierarchy of classes that classify living organisms using observed characteristics. Within the planetary science domain, the taxonomy of data granules includes images, time/series, spectra, tables, and cubes. Internet search engine taxonomies are increasingly important for classifying and organizing the millions if not billions of electronic documents being served. Within object-oriented data model, an implicit taxonomy exists as the class hierarchy of modeled domain entities.

A **vocabulary** is a collection of domain terms and their definitions. This is often an informally collected list of terms used within the domain and can include acronyms and abbreviations.

A **data element dictionary** provides a more exhaustive description of domain terms and is a key component of a data model. It is a formally collected list of domain terms that have been defined using a set of standard attributes. These attributes include at least name and definition and optionally other attributes such as registration authority and valid value type. A data element dictionary supports the design and specification of application-oriented data models, databases, and message types for data interchange. It also supports the use of data in communications and information processing systems and is critical for interchanging or referencing among various collections of data elements.

ISO/IEC 11179 [9] is a framework for the specification and standardization of data elements. This specification is proposed as the standard reference for attributes to be used in data element dictionaries for the information architecture. As an international standard, it provides a common basis for data element definition and classification. The specification defines four data element attribute categories namely identifying, definitional, representational, and administrative as illustrated in Figure 3.

Copyright Jet Propulsion Laboratory, California Institute of Technology

ISO/IEC 11179 Implementation
Basic Attributes

Attribute category	Name of data element attribute	Obligation	Maximum size (characters)	
Identifying	- name	M	80	
	- identifier	M	80	
	- version	M	10	
	- registration authority	M	35	
	- synonymous name			
	- context	O	80	
		M	80	
Definitional	- definition	M	4000	
Representational	- datatype of data element values	M	35	
	- maximum size of data element values	M	15	
	- minimum size of data element values			
	- permissible data element values	M	15	
		M	80	
Administrative	- comments	O	4000	

Figure 3 – ISO/IEC 11179 Basic Attributes.

A **schema** specifies how the data elements are grouped to describe domain entities and how the entities are related. For example, the relational data model uses a schema to describe data elements as columns in a table and how the tables are related through primary keys. In object-based models, the data element dictionary and schema are more closely integrated since the object hierarchy implicitly associates data elements (attributes) with objects through class definition.

Finally, where information system data models typically deal with readily understood data items such as transaction amounts or camera exposure durations, an **ontology** deals with more abstract or conceptual entities, their existence, and relationships. For example, the planet Jupiter can be both the object of an astrophysics observation and the target of a planetary mission flyby. However, because the astrophysics and planetary domains model target objects differently, a more abstract target object with attributes acceptable to both domains can be defined and then the two domain specific target objects can be related through the abstract one.

Enabling interoperability across domains is strongly dependent on metadata compatibility. [10] Therefore ontologies hold the key to enabling interoperability since they provide the most comprehensive data models. As described above, planetary science

Copyright Jet Propulsion Laboratory, California Institute of Technology

and astrophysics both model target bodies. However, unless the identifiers used within the domains are mapped across the domains, applications will not be able to support data search much less interoperability across the domains. In fact as more sophisticated interoperability such as data transformation and correlation is requested, deeper levels of metadata mapping will be required. For example, once a target body is identified, sufficient metadata must be available for coordinate system conversion.

For the information architecture it is proposed that the reference data model adopted or developed include capabilities for specifying ontological concepts including the definition of conceptual objects and their relationships to domain specific data models. The ISO/IEC 11179 specification has been used for this purpose and will be considered. The reference model should also be comprehensive enough to allow the design of data models for domain vocabularies and taxonomies.

3. The JPL Namespace

In general, a namespace uniquely identifies a set of names so that there is no ambiguity when objects having different origins but the same names are mixed together. [1] Within the information architecture a namespace will typically be domain specific, provides a vocabulary for domain discourse, and is managed by a domain registration authority.

Many information architecture namespace issues will be addressed in the data element dictionary since the ISO/IEC 11179 specification allows the distinguishing of name sets using attributes such as registration authority and context.

At JPL, we have namespaces for earth science, planetary science, engineering and institutional disciplines. The Planetary Data System (PDS) and the STEP community define namespaces for planetary science and manufacturing engineering respectively. The Dublin Core Metadata Initiative [4] defines a namespace recommendation for describing electronic resources on the Internet.

Namespace issues are fundamental to information architecture development. We propose to identify and register the enterprise namespaces, understand the relationship between namespaces, and normalize namespace frameworks to allow namespace sharing, comparison, and integration. Figure 4 illustrates some example JPL namespace hierarchies.

[1] Whatis

Copyright Jet Propulsion Laboratory, California Institute of Technology

JPL Namespaces

- Earth Science
  - Atmospheres
  - Oceans
- Space Science (Planetary)
  - Atmospheres
  - Geosciences
  - Rings
- Administrative
  - Facilities
  - Financial
- Mission
  - Voyager
  - Galileo
  - MGS
- Engineering
  - Mechanical
  - Electrical
  - Chemical

Figure 4 – Example JPL Namespaces.

4. The JPL Resource Space

A resource space is a collection of information resources typically within a single domain. Examples of resources include an image from a spacecraft, an instrument design document, and the web service that serves the document. A resource can be classified as either a data resource (e.g. image) or a system resource (e.g. web service).

To describe a resource within an information architecture we propose to use a profile, an outline that includes the main attributes of the resource. A profile is implemented as a document and uses domain terminology to identify, describe, and provide the location of the resource. Profiles support the search and identification of resources within the system itself as well for system users. Profiles can also be used to associate data with services.

We propose to write the profile document using the eXtensible Markup Language (XML) [6] [7]. The advantages of using XML include superior expressiveness to the Hypertext Markup Language (HTML) [11], simplicity as compared to the Standard Generalized Markup Language (SGML) [12] in use and syntax, wide acceptance as an Electronic Data Interchange (EDI) standard, and most compelling, its flexibility. The flexibility of XML allows us to develop a generic structure for managing resource descriptions from any domain.

The Document Type Definition (DTD) specification in Figure 5 illustrates the three parts of a resource profile: the profile attributes, resource attributes, and the profile elements. We use a DTD specification because the technology is widely supported but are considering XML-Schema since it is now a recommendation of the World Wide Web Consortium (W3C).

Copyright Jet Propulsion Laboratory, California Institute of Technology

The profAttributes section of the XML document contains the attributes that describe the profile itself. For example, the profId attribute provides a system-wide unique identifier for a profile document and the profStatusId provides an indicator of the profile's current status.

```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT profiles
 (profile*)>

<!ELEMENT profile
 (profAttributes,
  resAttributes,
  profElement*)>

  <!ELEMENT profAttributes
   (profId, profVersion?, profType,
    profStatusId, profSecurityType?,
    profParentId?, profChildId*,
    profRegAuthority?, profRevisionNote*, profDataDictId?)>

  <!ELEMENT resAttributes
   (Identifier, Title?, Format*, Description?, Creator*, Subject*,
    Publisher*, Contributor*, Date*, Type*, Source*,
    Language*, Relation*, Coverage*, Rights*,
    resContext+, resAggregation?, resClass, resLocation*)>

  <!ELEMENT profElement
   (elemId?, elemName, elemDesc?, elemType?, elemUnit?,
    elemEnumFlag*, (elemValue* |
       (elemMinValue, elemMaxValue)),
    elemSynonym*,
    elemObligation?, elemMaxOccurrence?, elemComment?)>
```

Figure 4 – Profile DTD

The resAttributes section of the XML document generically describes the resource using the 15 recommended elements of the Dublin Core initiative. Three additional attributes beyond the Dublin Core attributes have been added. These are resContext, which identifies the application environment or discipline from which the resource originates and is derived from a domain taxonomy. For example, NASA.PDS.GEOSCIENCE is used to indicate that the resource is associated with the Geoscience node of NASA's Planetary Data System. The resLocation provides the location of the resource, typically represented as a URL. Finally, resClass identifies the resource within a taxonomy of resource types. Examples resClass values are system.productServer, application.interface. data.granule, and data.dataSet.

The final section of the profile, the profile element section, uses domain terminology to describe the data content of the resource. For example, the Planetary Data System (PDS) maintains an inventory of all planetary science data sets. Within this inventory, the data sets are indexed on the identifiers of the instruments and target bodies involved in the observation. The profile of the inventory resource would include instrument and target

Copyright Jet Propulsion Laboratory, California Institute of Technology

body identifiers to indicate that the resource can resolve queries with instrument and target constraints. This is illustrated in Figure 5.

As also illustrated in Figure 5, each data element is defined using the metaattributes elemId, elemName, and elemValue. These metaattributes are consistent with those proposed as part of a JPL data element definition to allow for reference and validation.

```
<profile>
 <profAttributes>
  <profId>1.2.3.4...</profId>
  <profType>PROFILE</profType>
 <profStatusId>ACTIVE</ profStatusId >
 </profAttributes>
 <resAttributes>
  <Identifier>1.2.3.5...</Identifier>
  <Title>Viking Orbiter 1...</Title>
  <Format>text/html</Format>
  <Language>en</Language>
  <resContext>NASA.PDS</resContext>
  <resClass>data.dataset.</resClass>
  <resLocation>http://pdscat...</resLocation>
 </resAttributes>
  <profElement>
   <elemId>1.2.3.6...</elemId>
   <elemName>INSTRUMENT_N...</elemName>
   <elemType>ENUMERATION</elemType>
   <elemValue>VISUAL IMAG ...</elemValue>
  </profElement>
  <profElement>
   <elemId>1.2.3.7...</elemId>
   <elemName>TARGET_NAME</elemName>
   <elemType>ENUMERATION</elemType>
   <elemValue>MARS</elemValue>
   <elemValue>PHOBOS</elemValue>
   <elemSynonym>ADS.OBJECT_ID
            </elemSynonym>
  </profElement>
</profile>
```

Figure 5 - Example Profile

The creation of profiles is readily accomplished by extracting the necessary terminology from the domain's data dictionary. Domains that do not have a data dictionary have a serious problem remedied by developing a domain data model. Between domains, profiles promote interoperability by providing a common structure and interchange language within which to describe system resources.

5. System Modeling for Data Architectures and Engineering

The capture of models to describe system interfaces and data products allows for the generation of interoperable data systems. Defining and capturing the models allows for engineers to understand relationships between systems. At JPL, we have many systems

Copyright Jet Propulsion Laboratory, California Institute of Technology

that both produce and consume data. Frequently, the definitions for data associated with these systems is not captured resulting in different interpretations by users of the data, and by engineers sustaining and maintaining the information system. In addition, engineers are often led to develop new systems that have data of similar content leading to confusion in the enterprise as registration authorities[2] becomes blurred.

In September 1987 John Zachman published a paper on information system architecture that is commonly referred to as the "Zachman Framework" [14]. The value of the Zachman Framework is that it provides enterprises with a process for describing and modeling information system architectures. "The Framework as it applies to Enterprises is simply a logical structure for classifying and organizing the descriptive representations of an Enterprise that are significant to the management of the Enterprise as well as to the development of the Enterprise's systems." The Zachman Framework has been identified as a two dimensional matrix for describing information system architectures. These dimensions encompass the roles in designing systems (planner, owner, designer and builder) and the data functions applied to the enterprise (who, what, where, why) respectively. Figure 6 shows the Zachman architecture applied to enterprise information architectures.

	Data (What)	**Function (How)**	**Network (Where)**	**People (Who)**	**Time (When)**	**Motivation (Why)**
Ballpark View	List of things important to the enterprise	List of processes the enterprise performs	List of locations where the enterprise operates	List of organizational units	List of business events/cycles	List of business goals/ strategies
Owner's View	Entity relationship diagram	Business process model	Logistics network (nodes and links)	Organization chart	Business master schedule	Business plan
Architect's View	Data model	Essential Data flow diagram; application architecture	Distributed system architecture	Human interface architecture	Dependency diagram, entity life history	Business rule model
Designer's View	Data architecture	System design: pseudo-code	System architecture	User interface	"Control flow" diagram	Business rule design
Builder's View	Data design, physical storage design	Detailed program design	Network architecture	Screens, User Access	Timing definitions	Rule specification in program logic
Functional System	Converted data	Executable programs	Communication	Trained People	Business events	Enforced Rules

Figure 6– The Zachman Framework

The Zachman framework provides a useful tool for looking across organizational boundaries and considering the relationships among systems, organizations, processes

[2] Registration authority defines the organization or entity that has responsibility for defining and managing an attribute or data element that exists within a JPL name space

Copyright Jet Propulsion Laboratory, California Institute of Technology

and people. As we consider the information architecture, we can use Zachman's architecture to define the various "views" of data, and how it is used across the enterprise. One key to addressing system interoperability, is defining explicit metamodels for describing JPL information systems. Metadata descriptions for systems is a critical step in the data engineering of building information systems. For example, if an attribute of a system is named "username," it is very difficult to understand its relationship to other information systems. Many information systems at JPL commonly use the term username, but have local implementations and definitions. Having a well defined metadata model that explicitly defines a data element named "username" allows for the definition of the namespace and resource which has control authority over that value.

An information architecture principle suggests three phases in the development of metadata for the purpose of enabling heterogeneous data system interoperability. The first phase is simply the identification of individual data system data dictionaries and the copying of the data dictionaries in their native representation into the metadata registry. This phase simply registers the data dictionaries as electronic resources and archives them for subsequent retrieval using the Dublin Core data elements.

The second phase of development involves the transforming of the data dictionary's native representation into a standard representation based on the ISO/IEC 11179 specification. This standard representation is then ingested into the data element registry. The resultant data element registry affords a first order level of interoperability by simply allowing data elements to be referenced by users and systems across domains.

The third phase of metadata development involves the analysis and comparison of data elements both within and across domains to identify or derive Common Data Elements (CDEs). This typically involves identifying similar element names or definitions, determining core concepts, possibly generalizing the concept, and determining key names and aliases. The resultant discipline, application, or enterprise specific CDEs enable higher levels of data system interoperability once adopted by individual data systems.

Understanding the relationships among information systems is critical to an enterprise like JPL. Developing rich, well defined software interfaces will simplify the construction of information systems with. Following a phased approach that captures data dictionaries, describes data elements using standard data element definitions, and develops relationships between data elements will be important to understanding how information systems relate across the various namespaces at JPL.

6. Data Policies and Standards

Copyright Jet Propulsion Laboratory, California Institute of Technology

The benefit of building an enterprise wide information architecture is highly dependent on implementation of consistent data policies and standards that can leverage the foundational architectural principles. Data standards are critical to improving software quality, and creating interoperability and software reuse, while data policies are critical to infusing common information architectural themes into information systems across the institution.

Standards exist at many levels. Industry standards organizations like the ISO/IEC produce industry-wide standards adopted far and wide. Discipline-centric organizations like the Consultative Committee of Space Data Systems produce standards typically adopted across operations within a discipline. Operation-centric organizations like JPL produce standards typically adopted within and sometimes outside of the organization that produced them. Information architecture at JPL needs to work at all three levels. First, we need to identify and recommend adoption of data standards from external standards bodies. Second, we need to participate and advocate data standards that affect JPL's primary businesses. And third, we need to develop standards that achieve economies of scale in building information systems across the institution. Currently, a great deal of value is lost in building enterprise applications and systems since there is very little adoption of common approaches and standards. The establishment of common models and dictionaries and will help to yield solutions which allow for consistent capture, location, access, exchange, interpretation, and distribution of data.

While developing data standards is a critical component of developing an information architecture, ensuring implementation of those standards requires policies and the support of JPL management. Redundancy and inconsistency in databases across JPL can be largely attributed to the lack of coordination among projects. Adoption of common models, dictionaries, and interfaces will reduce information system complexity and large-scale integration efforts—but only so long as common infrastructure definitions can be identified and defined. This allows for vertically developed systems to take a horizontal approach to integrating common infrastructure pieces. In order to move to horizontally integrated information systems, JPL needs a central group that works across the organization to develop information architectural standards. The information architecture team, under the Chief Information Officer, needs to develop standards and architectures which enable the construction of enterprise applications and systems, while guiding information architecture efforts in different disciplines. Information architectural concepts may be similar between engineering and finance, however, the actual implementation of models, dictionaries and interfaces will most likely be very different. As such, the information architecture team will need the support of the CIO's office to set policy and guidelines for projects building information systems, and will need to make sure projects implement the standards and coordinate their data management activities—otherwise the value of defining information architecture at the enterprise level will be lost.

Copyright Jet Propulsion Laboratory, California Institute of Technology

7. Data Processes and Functions

Data processes allow the management of individual data granules and their meaning, behavior, and relationships. The information architecture contains data processes that are consistent within the JPL data space and across namespaces. It allows for the construction and management of the data resources.

7.1 Capture

Data capture is the process of collecting, organizing, describing, managing and preserving data in a durable form. Data modeling formalizes the data organizing principles to be used in capturing data. Data capture requires a standard modeling language, standard modeling tools, data modeling expertise and standard model repositories. It also requires tools for the collection and validation of the data against the defined model.

7.2 Locate

Data location is the process of locating data assets within a specific data space. Within the resource space, data and system resources are distributed. In order to navigate the resource space, a resource map needs to be defined that identifies systems their associated data resources in order to locate necessary data assets.

7.3 Access

Data access is the process of gaining the ability to query and retrieve data resources. Data owners control access privileges for authenticating and authorizing access to system and data resources. It is recommended that JPL data resources use standard protocols for data access where possible. For example, the Open Database Connectivity API (ODBC), The Java Database Connectivity API (JDBC), and the Lightweight Directory Access Protocol (LDAP) are widely accepted standards for accessing relational and directory databases.

7.4 Exchange

Data exchange is the process of exchanging data between two information providers. It is described by a common metamodel and supports standard data formats. Data exchange

Copyright Jet Propulsion Laboratory, California Institute of Technology

is supported by both a transport layer and the data description layer. There are several frameworks for exchanging data that exist using protocols. We recommend the use of XML as the data description layer. In fact, separating the transport layer from the data description layer allows for the adoption of various computing frameworks and middleware. Transport layers can include HTTP, IIOP, SOAP, etc.

7.5 Interpret

Data interpretation is the process of providing meaning to the data given the content, domain, and semantics associated with the data. It uses the underlying data model and associated data dictionary in order to provide context for interpreting the data. Data interpretation is a key function of the information architecture in that all clients of an information architecture need to be able to interpret data, and apply the associated model defined by its presence in a particular resource, name and data space.

7.6 Distribute

Data distribution focuses on the distribution of data products to information users. It can include the packaging of data granules in order to create information packages for distribution. The package includes the construct of both data and metadata. It can also include ancillary data, which is important in order to create the product. Constructing a package within the data space may require access and aggregation of disparate resources across the resource space.

8. Data Management Services

The Enterprise Data Management (EDM) services provide the information technology infrastructure to support the information architecture processes. The following table provides the mapping between the processes and the EDM services.

Process	**EDM Service**
Capture	Catalog and Archive Service, Database Hosting
Locate	Metadata Service: Resource Registry
Access	Product Exchange Service
Exchange	Product Exchange Service
Interpret	Metadata Service: Metadata Registry
Distribute	Data Distribution Service

Copyright Jet Propulsion Laboratory, California Institute of Technology

8.1 Database Hosting Service

The Data Access Service (DTA) provides a database hosting service chartered to achieve a reliable, cost effective, and responsive database hosting implementation for JPL projects. This allows DTA to reduce the overall cost of labor, hardware and software, and maintain a highly specialized database administration staff. The database hosting service is available laboratory wide and significantly reduces the startup costs, resources, and commitment a project needs to initially make to a data management solution.

Integrating institutional-grade applications with the Data Access Service addresses a critical component of an information architecture: data capture and access. The ability to gain access to and reliably capture data is primary for enabling enterprise-wide data management. Enterprise applications at JPL need the reliability and maintainability of an institutional solution that will insure that enterprise data is available twenty-four hours a day, seven days a week.

8.2 Metadata Service

The purpose of the metadata service is to capture and define the resource space at JPL. The metadata service provides a series of registries that allow for the management of data dictionaries, data elements, and resource descriptions. This service is designed around three major capabilities: capturing data dictionary schemas, capturing data elements, and capturing resource definitions for data systems, data sets, and data products. Using this registry, data and users can browse the data elements used by each data system as well as query the registry through a program interface.

The resource registry, the second major component of the metadata service manages metadata descriptions of resources across and within distributed data systems. The metadata descriptions are created from and validated against the metadata registry and are packaged as profiles. Resources can be almost any system entity including subsystems, services, data volumes, data collections, or data items. Examples of space science data items include images, spectra, and documents.

8.3 Catalog and Archive Service

The purpose of this service is to provide an active product storage and retrieval capability for missions and projects. The service is designed around these major capabilities:

Copyright Jet Propulsion Laboratory, California Institute of Technology

- Distributed access utilizing an Application Program Interface (API)
- Product type flexibility (archiving any BLOB)
- Cataloging based on data elements (from Metadata Service)
- Product-specific task execution (specialized for instantiations based on project need)

8.4 Product Exchange Service

The product exchange service implements the conceptual information architecture that enables interchange between applications residing on the same or different host. The product exchange service provides an extension on top of basic messaging which enables applications to send queries to gain access to distributed resources. The product exchange service requires a standard interface specification be supported in order for two information systems to exchange information. The specification means that both systems must agree on both the structure and content. The structure requires that both systems receive the information in the expected format. The content requires that both systems have an agreed upon model for expressing the data that is being shared. Often this expression centers on a metadata model. This requires a taxonomy for communication that is consistent with the data dictionary registry as defined in the Metadata Service.

XML is continuing to gain in popularity as a means for managing the structure for information interchange. While there is currently a significant amount of hype regarding XML as the "silver bullet" necessary to solve all the integration problems, there are some advantages that XML is providing to industry. In particular, XML is a specification language. It does not provide the means to transfer data between two particular network-based entities. This means that the data content is abstracted above the transport mechanism. This allows for any of the technologies supporting the transport to evolve independent of the specification language. This is important since the language needs to be persistent.

Currently there are several efforts underway to define and develop interchange mechanisms using XML. One such effort is SOAP (Simple Object Access Protocol) championed by Microsoft to provide a way to leverage off the Internet infrastructure that has already been deployed. SOAP allows for web services to exchange information using HTTP.

8.5 Data Distribution Service

Data distribution is becoming an important information technology issue because of increasing volumes of data and the increasing physical distribution of data sets. Data distribution will involve packaging and release of archived data products on hard media,

Copyright Jet Propulsion Laboratory, California Institute of Technology

and electronically. These services will integrate with the Data Hosting Service, the Catalog and Archive Service and the Institutional Security Service.

9. Benefits to JPL

JPL produces an enormous amount of data. In fact, data is JPL's major commodity. It is used to define requirements, capture designs, command spacecraft, collaborate with team members, capture and distribute scientific products, communicate with the public, and manage JPL's business systems. The following section outlines benefits for many key areas at JPL.

9.1 JPL Mission Information Systems

A key thrust of information systems at JPL is enabling the capture and distribution of data acquired throughout the lifecycle of a mission. JPL plays in an important role in providing the infrastructure to support the flow of data for a mission. This includes the need to move data across multiple heterogeneous systems. Missions are becoming more and more distributed, as a result, we there is a need to have a solid information model that describes the data being captured. Specifically, having information models that describe the data objects would allow for correct interpretation and common interfaces to be developed which could use these information models when exchanging information.

Organizations such as the Consultative Committee on Space Science Data Systems (CCSDS) has been defining data management standards for the exchange of information. However, CCSDS has not defined common information models for describing the information products that have been captured. This is a key in laying the infrastructure for collaborative mission management across multiple institutions. Providing accurate interpretation and processing of the data is essential to preserving the quality of the results from any mission.

9.2 JPL Earth Science Data Systems

Earth Science missions is another area that could benefit from information architecture and common data management services. With the new thrust within NASA to launch more missions faster, better and cheaper, JPL can no longer afford to repeatedly re-implement common data processing requirements. One such common requirement for a JPL instrument flying aboard an Earth Orbiting Satellite is a ground data processing system for processing, managing and distributing in near-real time the high volumes of science data returned by that instrument.

The common process management functions comprising a successful data processing system are product and metadata management, data processing control and data distribution. While the large, heavily funded missions of the past have had both the time

Copyright Jet Propulsion Laboratory, California Institute of Technology

and money needed to develop their own mission-specific data processing system from the ground up, such large-scale missions are no longer supported by either the NASA vision or budget. Consequently, it is imperative that Earth Science Data Systems be able to focus the majority of their budget on the design and development of the software required to support the scientific goal of the mission. For this to be possible, a flexible and reliable framework providing the common data processing functions is required into which mission-specific software can easily be "plugged in".

One example of the use of such a framework is demonstrated in the partnership between the Enterprise Data Architecture (EDA) task and the SeaWinds projects. The SeaWinds on ADEOSII mission is scheduled to launch in November 2002 as a follow-on to the SeaWinds on QuikSCAT mission. QuikSCAT was launched in June of 1999 and has been successfully returning science data for nearly three years. The process management functions of the science data processing system for QuikSCAT is based on a client-server architecture which employs the Sybase Open Server libraries for process-to-process communication. Though the current science data processing system is functioning successfully, the process management functions are database-dependent, platform-dependent and operator-intensive. With the collaboration between SeaWinds and EDA, the science data processing system for SeaWinds on ADEOSII is based on a framework that comprises components as building blocks for constructing future science data processing systems that are database-independent, platform-independent and capable of "lights out" operation.

The flexibility, reliability and scalability of the framework for SeaWinds on ADEOSII has so many benefits that a decision was made to retro-fit the new framework for ADEOSII back in to the currently operating QuikSCAT science data processing system. The implementation for the initial phase of the retro-fit was accomplished in two weeks.

9.3 JPL Engineering Data Systems

The engineering disciplines at JPL have long been interested in building collaborative environments by interconnecting tool and environments and facilitating an integrated data management approach for mission conception to launch. Included is collecting data to analyze decisions and perform trade studies between certain decisions addressing both risk and cost, as well as the ability for the mission to meet its objectives and requirements.

The Center for Space Mission Architecture and Design (CSMAD) has invested in building tool environments for capturing designs, however, very little interoperability between tools exist.

The Product Data Management System (PDMS) has been focusing on the capture of system design for long term preservation and archiving purposes.

Copyright Jet Propulsion Laboratory, California Institute of Technology

Team-X initiatives have been used to leverage models in order to analyze mission development costs and engineering decisisions, yet sharing information between Team-X studies can certainly be difficult.

The Information Architecture concept proposed in this white paper directly addresses an architecture with substantial benefit to JPL's engineering environment. It provides an architecture for sharing information between tools leveraging a collaborative environment to support engineering, and it provides the ability to share information between phases of the design process. It describes basic approaches to data modeling allows for data to be captured and useful long term. It describes common methods for working with the data and uses industry adopted approaches for describing and working with the data through such data languages as XML. An integrated model for engineering will also be useful as we look for ways to understand how to leverage data across domains including tying the financial data into the engineering design.

9.4 NASA Planetary Data System

NASA's Planetary Data System (PDS) is an active science data archive managed by scientists for NASA's planetary science community. In operation since 1990, the archive currently has about five terabytes of data covering 30 years of solar system exploration. This archive will double in size within a year and will increase by a factor of 60 within three years. The system is distributed, consisting of several science disciplines nodes, support nodes and a central node. Currently the primary means of data distribution is through the creation of multiple copies of CD/DVD media and their distribution to NASA funded scientists. Some online access is allowed to subsets of the archive at the discipline nodes.

With the dramatic increase in size of instrument data sets, the creation and distribution of data on physical media is no longer viable. Components of the information architecture proposed in this paper provide an excellent means of addressing this problem. The key to the solution is the wealth of metadata contained within the archive.

The approach, currently being prototyped, is to overlay a multi-tiered, component-based systems architecture that operates concomitantly with existing subsystems of the PDS. Using distributed processing and a message-based communications protocol, existing subsystems will be encapsulated as web services. For example, data repositories such as disk farms will be provided with product servers, allowing product retrieval through a common interface. Similarly existing catalogs will provided with profile servers, also allowing product search through a common interface. The use of XML will provide a common interchange language. Most importantly, PDS subsystems will remain geographically distributed and locally managed.

Copyright Jet Propulsion Laboratory, California Institute of Technology

9.5 JPL Document Management Services

The Information Architecture proposed in this paper provides a great opportunity for document management at JPL. JPL generates a large number of documents across projects which exist in a variety of information repositories, however, there is very little interoperability between documents. The information architecture for document management needs both a document and system architecture that defines the syntactic, semantic, and behavioral models that allows users to leverage the information processes defined in this paper.

Users need the ability to capture documents in interoperable repositories. JPL has operational systems for storing documents including JPL Rules, DocuShare, and the Product Data Management System (PDMS). However, the semantic models that describe these documents, along with the syntactic models that define the document structures are very minimally defined and enforced by each repository. There are no adherence to common models for describing the document products which makes interpreting and relating documents a challenge.

It is also difficult for general users at JPL to locate documents since they need to understand the particulars of each system implementation. The lack of a common model makes it difficult to have both a breadth and depth search of institutional repositories. A common model which defines both the metadata associated with a document, as well as common definitions of document structures would simplify the process of locating and interpreting documents. Common definition of document structures would allow for not only locating documents, but parsing and interpreting documents themselves. Common structures for engineering documents, memorandums and policies are just a few documents that could be standardized from a structural standpoint.

In addition to capturing, locating, and interpreting documents is accessing the document repositories that store these documents. First, having knowledge of where documents are located is a critical need to unifying the document space across the instition. Second, defining and publishing public interfaces to these document systems is critical to being able to develop access methods across institutional document systems.

The JPL Data Management services will provide the basic infrastructure for cataloging, archiving, accessing, and exchanging documents across the institution. Implementing common document models that use these services will begin to form an integrated document architecture which will reduce the time for users to find documents, increase the usability of documents, and preserve critical information resources long term.

9.6 JPL Information Services

Copyright Jet Propulsion Laboratory, California Institute of Technology

The Institutional Computing and Information Services (ICIS) is the premier provider at JPL for information services. It provides services to end users and projects, providing the basic information technology infrastructure for operating the laboratories information systems. The JPL information services needs to move to a level of interoperability that currently does not exist. This includes building common provisioning and work flow models that facilitates service management. Presently, the initiation, termination, and relationship between services and users is inconsistently implemented. This makes it difficult for both users and services providers to consistently and efficiently manage the service experience.

In addition, the way in which subscriptions are passed to the Unified Charging System is inconsistent. The methods to capture and subscribe users to services is ad hoc across ICIS from service initiation to service termination. This leads to incomplete and inconsistent data management practices, and ultimately increased costs and reduced customer satisfaction as it provides opportunities for errors to be introduced.

The information architecture processes defined in this paper, if implemented for JPL Information Services, would greatly enhance the ability for ICIS to perform service data management. This includes consistently capturing information about users and associated service subscriptions, passing service requests between all ICIS services, and abstracting users away from having to understand which groups within ICIS provides what services. Common user interfaces built on a common infrastructure should exist which present a unified view of all ICIS services, similar to the way the Unified Charging System (UCS) currently works. Implementing the Information Architecture within ICIS would ultimately lead to reduced operation and sustaining costs, and greater customer satisfaction since service implementations would be predictable, and users would not have to understand each services unique implementation.

9.7 JPL Business Systems

JPL Business Systems is a principal source for the distribution of information regarding both personnel and laboratory finances. As such, many systems at the laboratory either directly or indirectly have interfaces with the JPL New Business System (NBS). A key benefit from the information architecture proposed in this white paper is to capture the model that describes the data that NBS generates and distributes. This includes providing a rich model of data elements that describe people (i.e. fullname, badge number, phone number, etc) and financial data information. Defining the structure and the content for the data that is distributed will be useful in building enterprise applications since data that is reused can be traced back to the source. It also allows users to correctly interpret data elements that are used. Another example is (ITAR). Currently, systems need to understand whether or not a user is a foreign national. Data attributes specifying this are often generated locally by an application, or are misused. Both centralizing and defining

Copyright Jet Propulsion Laboratory, California Institute of Technology

these characteristics will have enormous benefit long term since these models for the basis for building institutional security and directory systems.

10. Plans and Recommendations

We believe that information architecture is critical to enabling JPL to meet its challenging objectives, and believe that it should be coordinated as a major initiative under the JPL CIO's office. There are many institutional projects at JPL that produce data, however, very little coordination exists between these projects. A key to reducing costs and increasing data reuse, is coordinating implementation efforts across the institution. The Institutional Computing and Information Services Office is responsible for providing and building institutional systems at JPL. As such, we believe that all software system development at the institutional level should be coordinated with the CIO's office in an effort to better coordinate the architectural approaches to consume and produce data products, improve the software and data quality produced by JPL information systems, and reduce data redundancy.

The information architecture team has identified several initiatives in order to define an information architecture at the institutional level. This includes analysis and modeling of the currently as built JPL information environment, identification of key standards to enable an interoperable information architecture, identifying opportunities for infusion of information architecture principles in existing systems, as well as describing information architectures for future JPL data systems.

The following process outlines the approach to define and implement an institutional information architecture:

- Definition of the existing architecture(s)
    - Definition of the JPL namespace
    - Identification of existing resources
    - Definition and capture of data dictionaries, taxonomies and vocabularies to describe information resources
    - Definition and capture of interface formats for data systems
- Establish an information architecture council within each namespace to support a common information model for that namespace
    - Establish and manage common data elements (CDEs)
    - Identification of needed institutional services for data management
    - Point-of-contact for understanding how to relate data within the namespace
- Establishment of enterprise information standards
    - Common language via XML
    - Data interchange standards

Copyright Jet Propulsion Laboratory, California Institute of Technology

    - Data dictionary standards
- Establish relationships across namespace boundaries
    - With application owners to create new applications
    - With Knowledge Management

Using the data and information modeling approaches described in this paper, we assert that capturing the information regarding data dictionaries, data formats, and data interfaces will be invaluable to understanding the information architecture of the laboratory. In addition, it will be a necessary step to formulating a map of the various namespaces at JPL, as well as the data resource spaces within each namespace. Definition of both the name and resource space will be critical for enabling many different types of applications at JPL, helping to drive relationships within and between these resources at JPL. Therefore, understanding the information architecture will be part of a plan to enabling a knowledge management infrastructure.

Data abstraction has long been identified as an important design characteristic of complex information systems. Data abstraction allows the construction of software components that use common interfaces in order to plug systems together. Modern computer science theory suggests that "design patterns" can be created from components and can yield a reference architecture. XML plays an important role in helping to build such a reference architecture. Defining and controlling the data and semantic architecture is important for constructing information systems where the information architecture can evolve independent of the technical architecture. It will also define the data format, packaging and interchange mechanisms and provide a vendor-independent architecture with the ability to incorporate new technologies and system paradigms (e.g. files, objects, goals, autonomy in spacecraft, constellations, etc).

We envision that capturing XML schemas to define data system formats will be an essential part of enabling interoperability. JPL, as a member of the Consultative Committee for Space Data Systems (CCSDS), is working on developing standards that address the management and exchange of data related to mission information systems, from cradle to grave. The JPL Information Systems Standards Program is working to identify XML data formats and architectures for building such systems. We envision that capturing this information across namespace boundaries is just as important, if not more, for building interoperable, well engineered information systems from an institutional perspective. It provides a language for creating a well-controlled information architecture, and a mechanism for refocusing systems implementations around data definitions and interfaces.

The institutional information architecture team will publish two documents on the JPL information architecture in fiscal year 2002. The first, will be considered version 1.0 of the JPL Information Architecture, and will consist of the overall information architecture, based on this technical concept paper, along with specific applications to activities in the

Copyright Jet Propulsion Laboratory, California Institute of Technology

institutional and space science namespaces. Version 2.0 will extend the document to include applications to engineering and Earth science.

11. References

[1] A brief introduction to IEEE P1471, IEEE's Recommended Practice for Architectural Description, IEEE Architecture Working Group, April 1999, http://www.pithecanthropus.com/~awg.

[2] Planetary Science Data Dictionary Document, JPL D-7116, Rev. D, July 1996http://pds.jpl.nasa.gov/psdd.pdf,.

[3] The STEP Project, ISO 10303, http://www.nist.gov/sc4/www/stepdocs.htm. /

[4] Dublin Core Metadata Initiative. The Dublin Core Element Set Version 1.1, Dublin: DMCI, July 1999, http://dublincore.org/.

[5] Universal Modeling Language (UML) http://www.w3.org/TR/html4/.

[6] Extensible Markup Language (XML), W3C Recommendation, http://www.w3.org/TR/html4/.

[7] Tim Bray et al. Extensible Markup Language (XML) 1.0 (Second Edition), Cambridge: World Wide Web Consortium, 2000.

[8] XDF (eXtensible Data Format), http://www.w3.org/TR/html4/

[9] ISO/IEC 11179 - Specification and Standardization of Data Elements, Parts 1-6, ISO/IEC specification, http://www.iso.ch/iso.

[10] Hughes, J. S., et al. "A Multi-Discipline Metadata Registry for Science Interoperability," Santa Fe: Open Forum on Metadata Registries, ISO/IEC JTC1/SC32, Data Management and Interchange, 2000.

[11] HTML 4.01 Specification, W3C Recommendation 24 December 1999, http://www.w3.org/TR/html4/.

[12] A Gentle Introduction to SGML, W3C Recommendation, http://www.w3.org/TR/html4/.

[13] XML-SCHEMA, W3C Recommendation http://www.w3.org/TR/html4/

[14] Zackman, John A. A Framework for Information Systems Architecture. IBM Systems Journal, vol. 26, no. 3, 1987. IBM Publication G321-5298. 914-945-3836 or 914-945-2018.

[15] DocBook. http://www.docbook.org.

[16] Reference Model for an Open Archival Information System (OAIS). Red Book. Issue 2. June 2001, http://www.docbook.org

[17] Crichton, D., et al. Enterprise Data Architecture White Paper. D-19947. Jet Propulsion Laboratory. November 2000.

[18] Foster, I., Kesselman, C. The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann Publishers, 1999.

[19] IEEE Glossary of Software Engineering Terminology, 610.12-1990.

[20] Cook, Melissa A. Building Enterprise Information Architectures. Prentice Hall, Inc., 1996.

Copyright Jet Propulsion Laboratory, California Institute of Technology